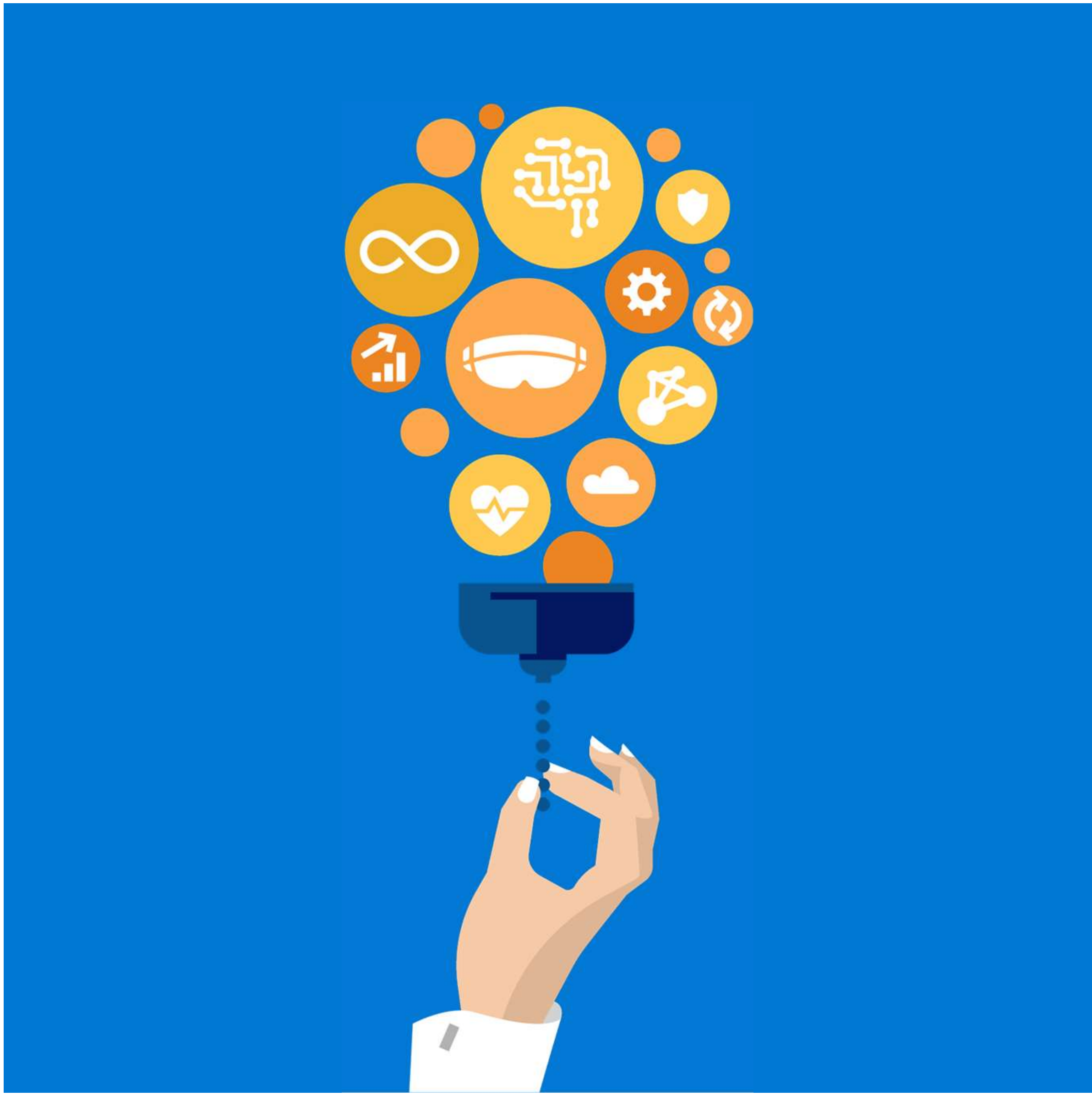
Microsoft

Deep Learning Vector Search Service

Jeffrey Zhu, Program Manager

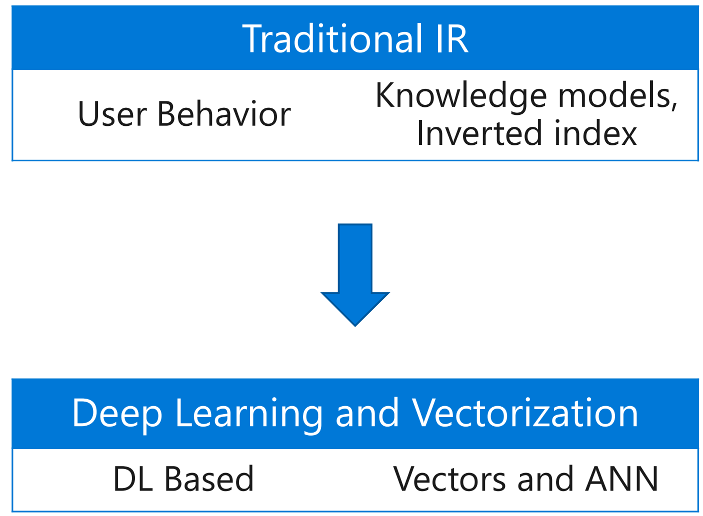
OpML '19



Evolution of Search

Classic information retrieval is based on keyword matches and user behavior signals

- Query rewrite and other alteration techniques cannot enumerate all keyword expansions
- Insufficient user signals for tail queries



Novel search scenarios have emerged

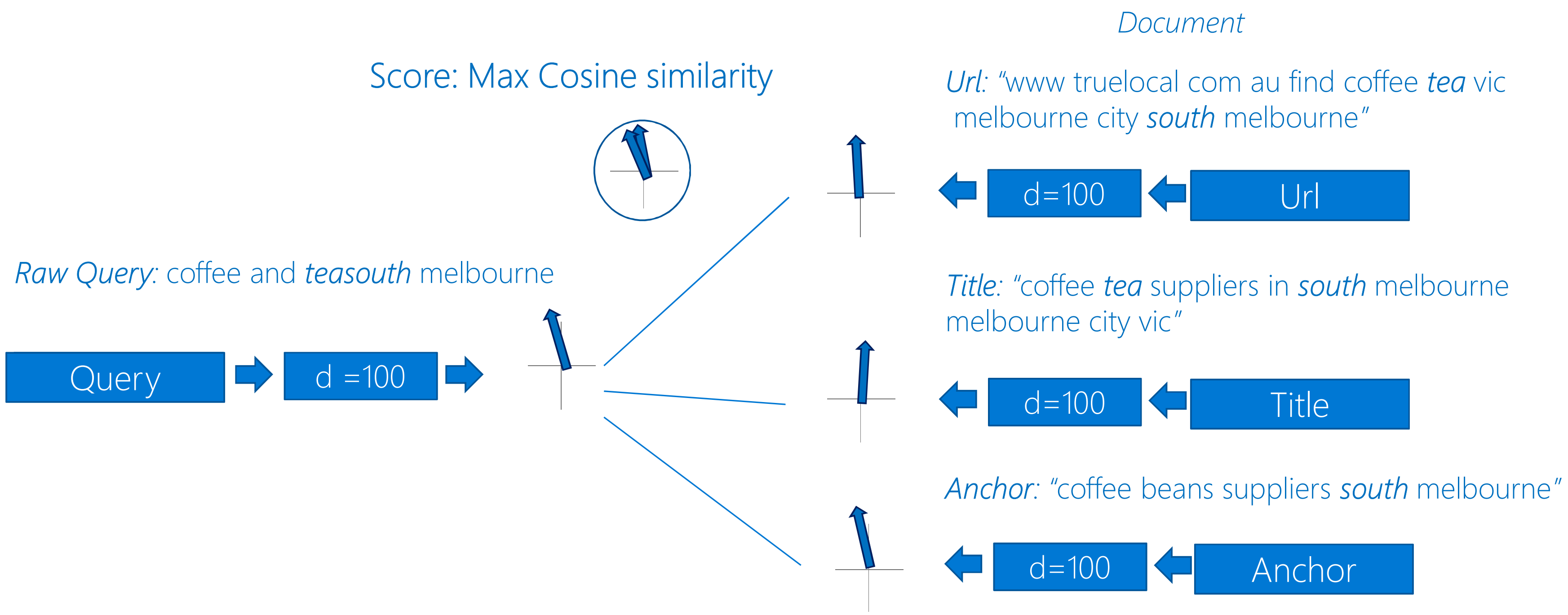
- Natural language/Conversation, Question and Answer, image/multimedia

Vector search is a critical technique to improve search and enabling new scenarios

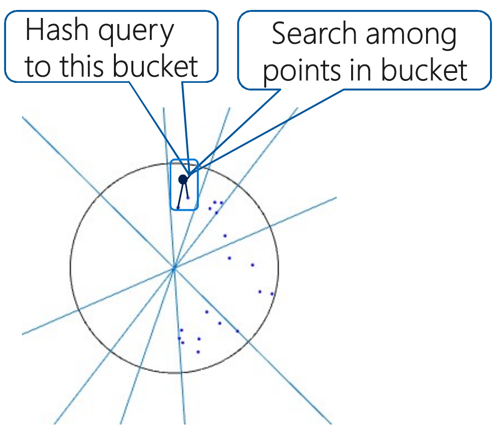
Content Vectorization

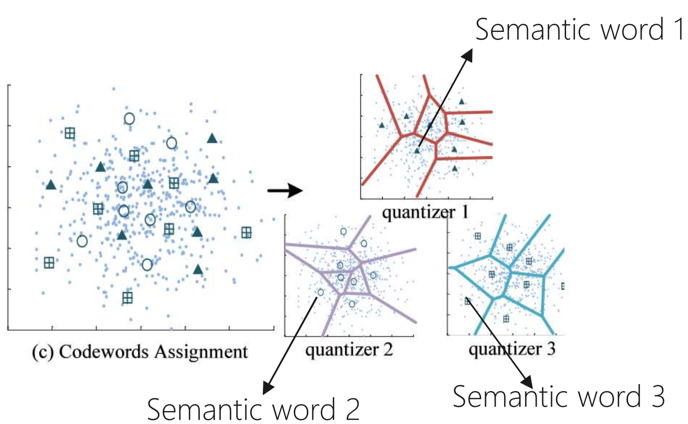
Use deep learning model to encode content as a vector

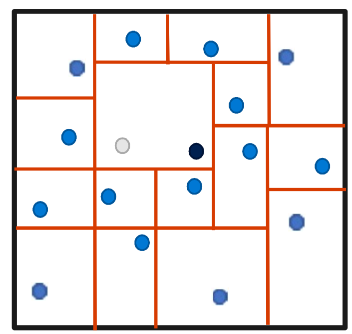
- Distance between vectors represents semantic similarity
- Better semantic representation, tolerant to out of vocabulary, spelling errors, connective words.



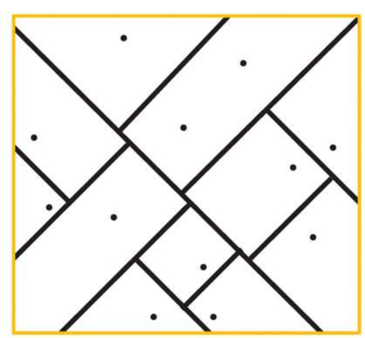
Vector Recall by Nearest Neighbor Search



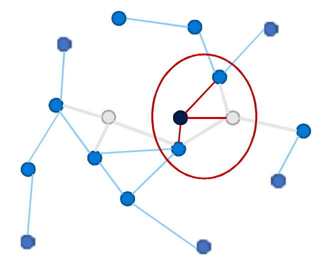




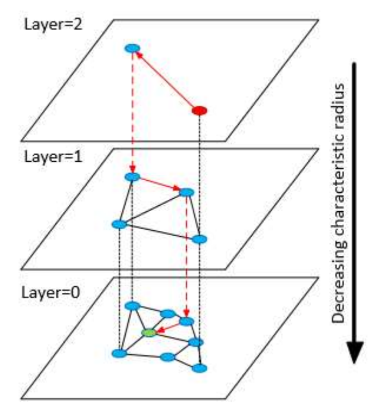
KD-tree



TP-tree

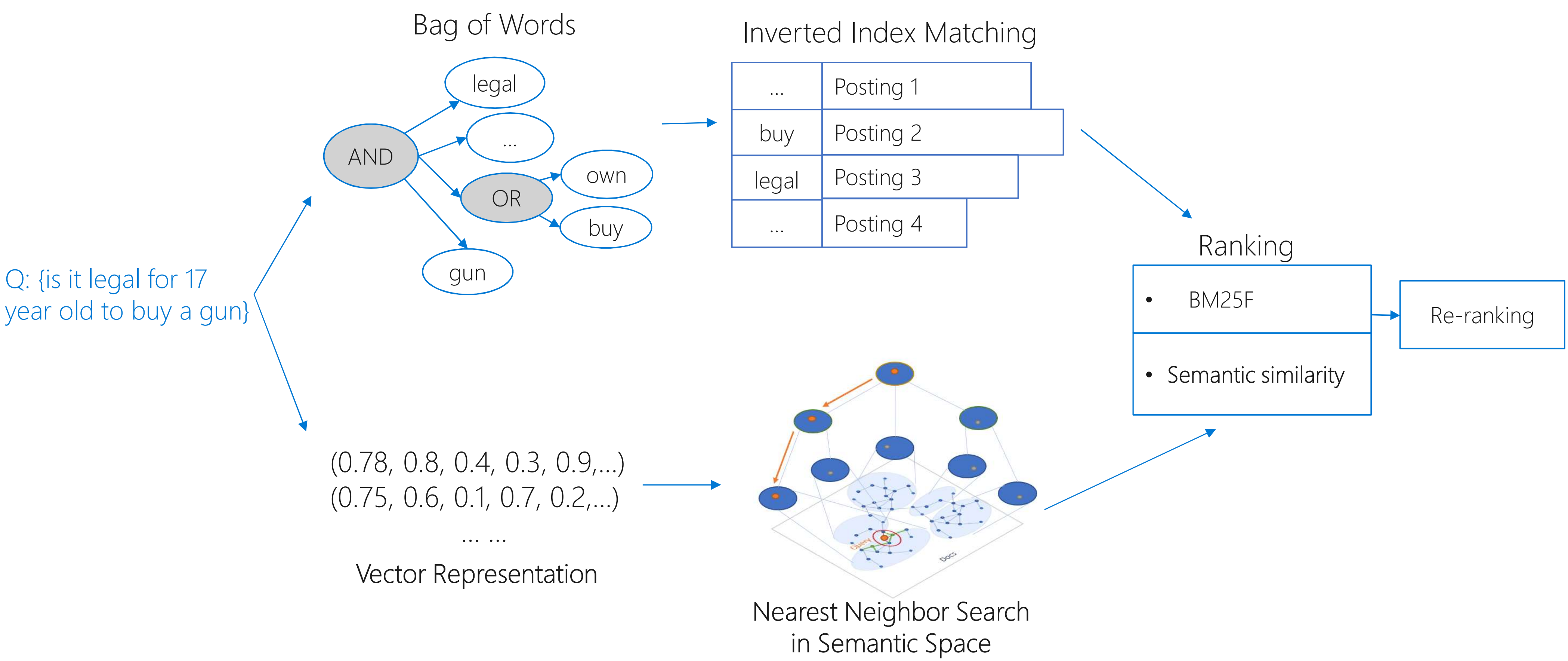


NNG



HNSW

From Keyword to Semantic Vector Search



SpaceV: Semantic Vector Search at Scale

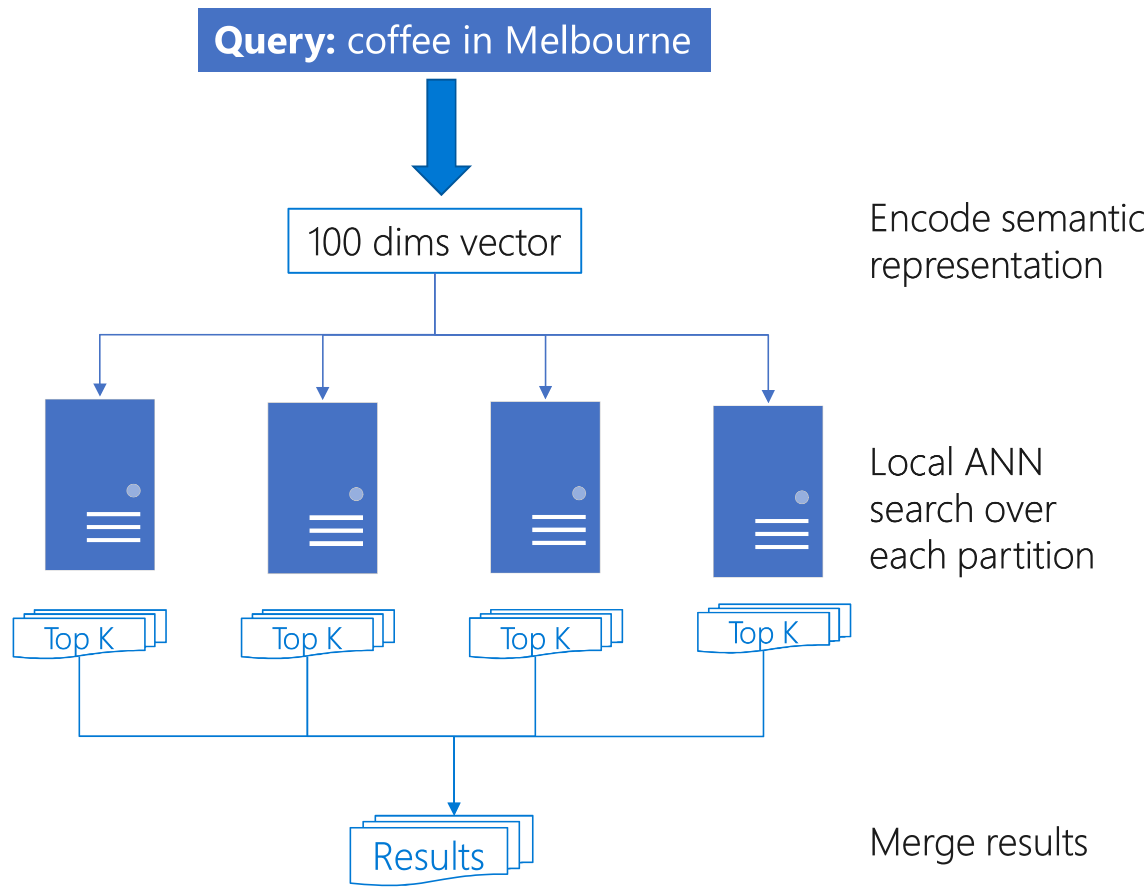
- Better fidelity (NCG@infinity) than keyword search + BM25F ranker with the same document sets
- Additional fidelity gain after combining with keyword search

L1 Fidelity on full index	Overall	Tail
Keyword + Vector Search	+3.24	+5.14

Deep Learning Vector Search Service

- Platform Capabilities
  - Performance: <10ms search latency
  - Scale: 100B+ vector index size
  - Agile: Fast experimentation + deploy
  - Flexible: Pluggable ANN algorithms

- Distributed serving
  - Randomly partition vectors into smaller vector indexes
  - Serving queries is distributed and aggregated before returning



SpaceV: Semantic Vector Search at Scale

- High scale and Low latency
  - 40B+ vectors
  - Served with N (N=3) replica in 500+ servers
  - High capacity: 240M vectors per machines * 1,800 QPS at most
  - Low latency: 5ms in average and 8ms in 95%ile

	QPS per replica	Avg latency (ms)	50% latency (ms)	95% latency (ms)
Normal Traffic	1,200	5.341	4.764	8.004
Peak Traffic	1,800	6.177	5.159	9.293

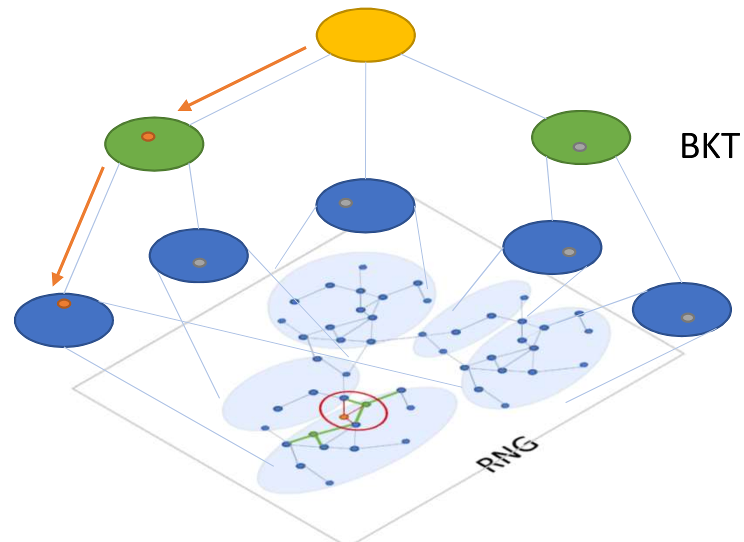
Key Innovations

- SPTAG - Approximate Nearest Neighbor Algorithm
  - Balanced k-means tree over relative neighbor graph
- Distributed Vector Index Serving
  - K-means clustering for distributed serving
- Lower Cost Serving Hardware
  - Offload index from memory to Solid State Disk (SSD)

SPTAG – Space Partition Tree and Graph

- Hybrid approach to achieve high recall for both low and high dimension vectors
  - BKT: Balanced K-means Tree
  - RNG: Relative Neighbor Graph

- Designed for efficiency, scale, and agility
  - Better trade-off between recall and latency
  - User customized distance
  - Incremental update



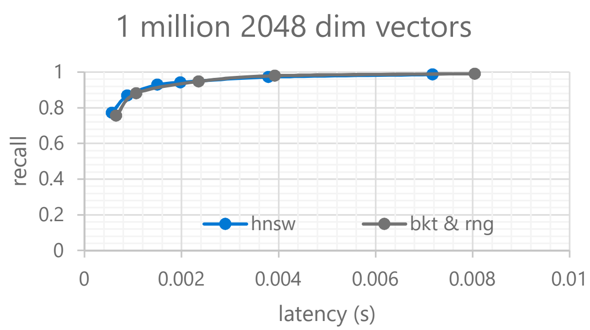
Balanced K-means Tree

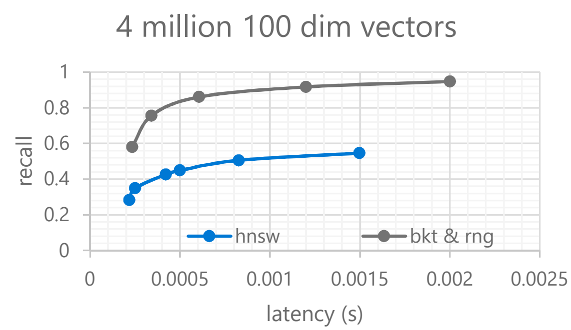
Object function: $\min_{H,C} ||X - HC||_F^2 + \lambda||\mathbf{1}^\top H||_2^2$

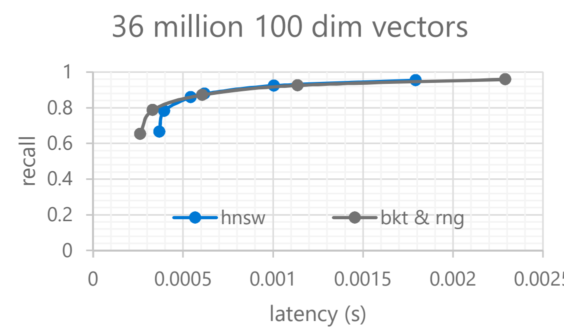
Cluster chosen: $k = \arg\min_i f(x_l, c_i) + \lambda s_i$

SPTAG – Space Partition Tree and Graph

- Evaluation
  - Three datasets: 1M 2048 dim, 4M 100 dim, 36M 100 dim
  - Two algorithms: HNSW, BKT & RNG



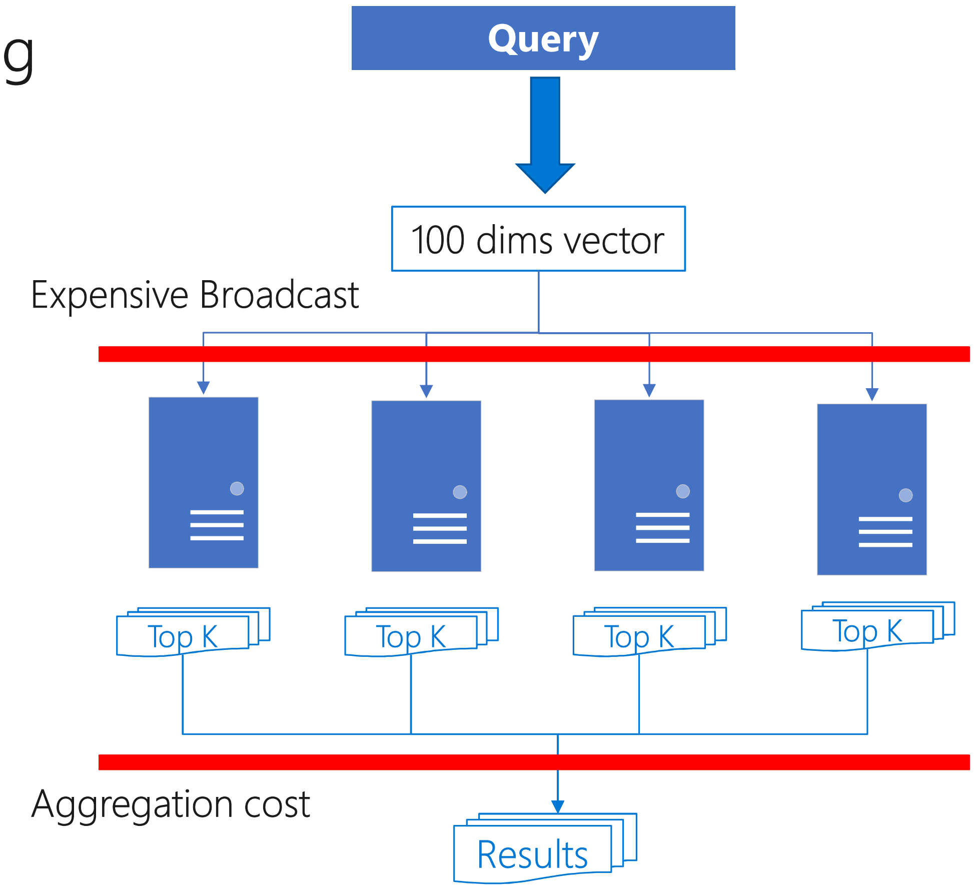




- Open source available at https://github.com/Microsoft/SPTAG

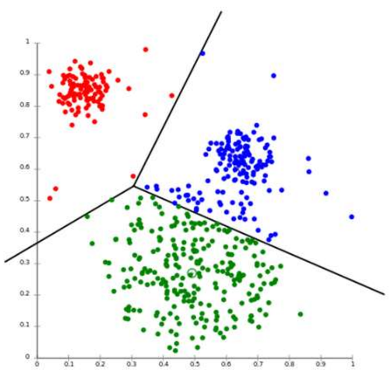
Distributed Vector Index Serving

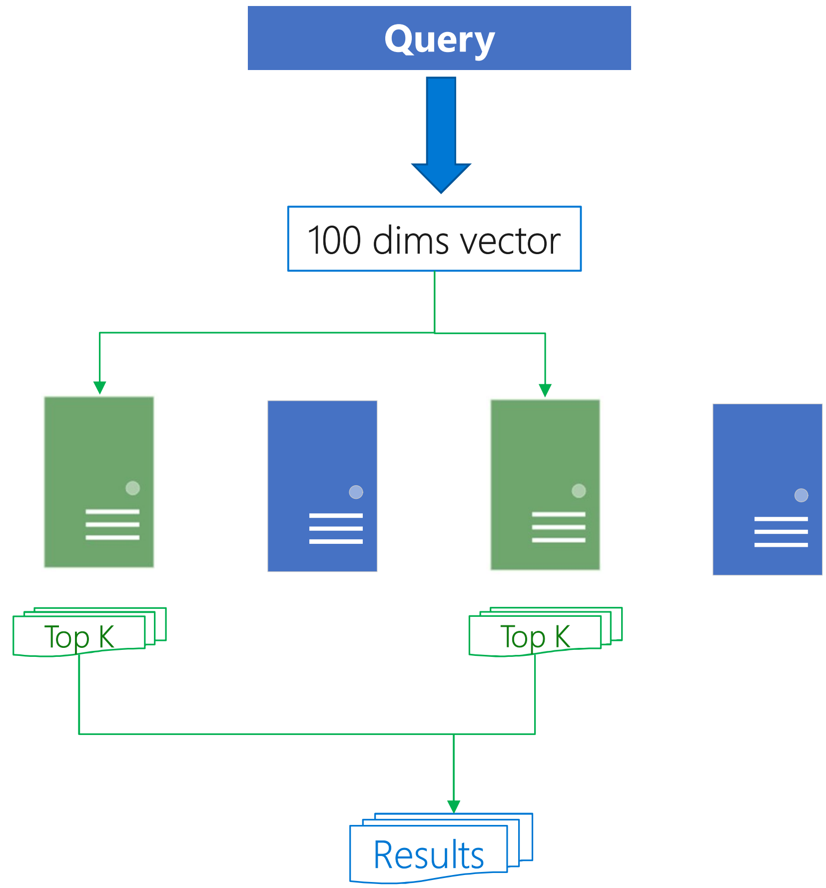
- Challenges with Distributed Serving
  - Poor scalability
  - Too much resource usage for each query
  - Poor latency – long tail



Distributed Vector Index Serving

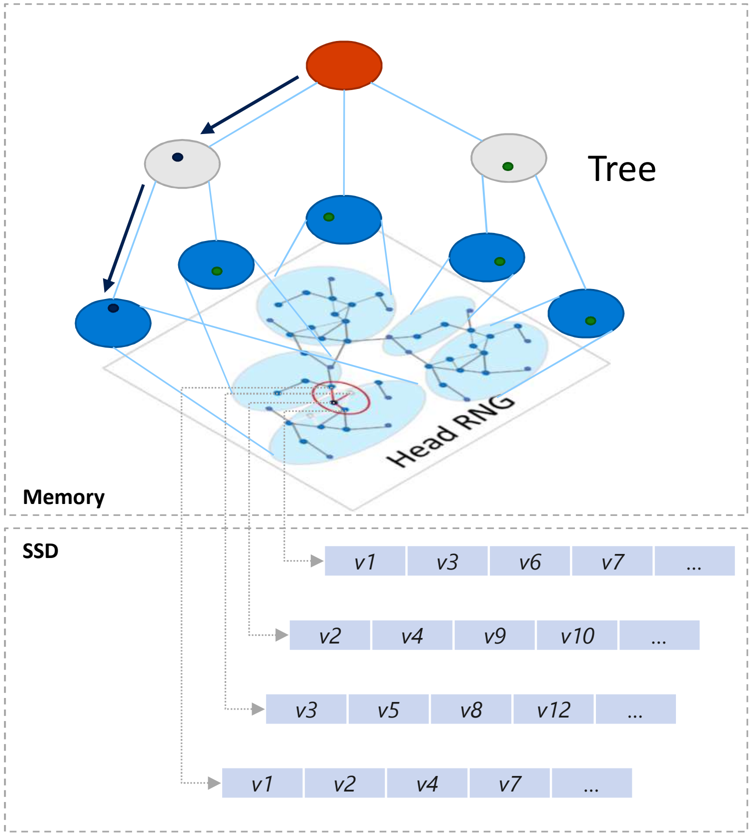
- Data partitioning with balanced k-means clustering
  - Each data partition maps to specific cluster
  - Each query is only sent to closest clusters (instead of global broadcast)
- Evaluation
  - Selecting top 5 out of 22 clusters can get the same recall as baseline, and only use 23% capacity





SSD Serving

- Challenges
  - Memory is bottleneck to lower cost serving
  - Memory cache hit rate is low due to ANN random access pattern
- ANN algorithm for SSD
  - Build head index from partial vector and serve in memory
  - Cluster tail vectors with head vectors as a center and serve in SSD



SSD Serving

- Evaluation
  - Dataset: 13 million 100 dim vectors
  - 67% memory saving

	Index Size	Metadata Size	In Memory	In SSD
Memory Serving	32.3G	6.6G	32.3G	-
SSD Serving	47.5G	6.6G	6.6G	40.9G

	Average	99%	Recall
Memory Serving	1.05ms	1.32ms	0.962
SSD Serving	3.07ms	5.90ms	0.929

Takeaways

- Vector search is a critical technique to improve web search and power new capabilities such as question and answering, image search, etc.
- Key innovations in ANN algorithm and distributed vector index serving allows DLVS platform to serve high scale vector search scenarios (100B+ vectors)
- Core ANN algorithm (SPTAG) is open source and available for developers to use
  - https://github.com/Microsoft/SPTAG